

Query 1:

- $\Pi_{CName, City}(Customer)$
- $\forall cx(Customer)$
 $\{ \langle cx.CName, cx.City \rangle : True \}$
- select CName, City
from Customer;

Query 2:

- $\Pi_{PName, Price}(\sigma_{(Producer='Apple' \vee Producer='Samsung') \wedge Year='2020'}(Part))$
- $\forall px(Part)$
 $\{ \langle px.PName, px.Price \rangle : (px.Producer='Apple' \vee px.Producer='Samsung') \wedge px.Year = '2020' \}$
- select PName, Price
from Part
where (Producer='Apple' or px.Producer='Samsung') and Year = '2020';

Query 3:

- $\Pi_{Date, Quantity, Amount}(\sigma_{(Producer='Apple' \wedge Year='2020')}(Part \bowtie Supply))$
- $\forall sx(Supply)$
 $\{ \langle sx.Date, sx.Quantity, sx.Amount \rangle :$
 $\quad \exists px(Part) (sx.P\# = px.P\# \wedge px.Producer='Apple' \wedge px.Year='2020') \}$
- select Date, Quantity, Amount
from Supply, Part
where Supply.P#=Part.P# and
Producer='Apple' and
Year='2020'

.

Query 4:

- $\Pi_{PName, CName, City}(\sigma_{(Quantity > 1000 \wedge Date > '2019-01-01')}(Customer \bowtie Supply \bowtie Part))$
- $\forall px(Part), \forall cx(Customer)$
 $\{ \langle px.PName, cx.CName, cx.City \rangle :$
 $\exists sx(Supply) (px.P\# = sx.P\# \wedge$
 $sx.C\# = cx.C\# \wedge$
 $px.Quantity > 1000 \wedge$
 $px.Date > '2019-01-01') \}$
- select PName, CName, City
 from Part, Customer, Supply
 where Part.P# = Supply.P# and
 Supply.C# = Customer.C# and
 Quantity > 1000 and
 Date > '2019-01-01'

Query 5:

- $\Pi_{CName, City}(\sigma_{Producer = 'Apple'} (Customer \bowtie Supply \bowtie Part))$
- $\forall cx(Customer)$
 $\{ \langle cx.CName, cx.City \rangle :$
 $\exists px(Part) \exists sx(Supply) (cx.C\# = sx.C\# \wedge$
 $sx.P\# = px.P\# \wedge$
 $px.Producer = 'Apple') \}$
- select distinct CName, City
 from Customer, Part, Supply
 where Customer.C# = Supply.C# and
 Supply.P# = Part.P# and
 Producer = 'Apple'

Query 6:

- $\Pi_{CName, City}(Customer) - \Pi_{CName, City}(\sigma_{Producer = 'Apple'} (Customer \bowtie Supply \bowtie Part))$
- $\forall cx(Customer)$
 $\{ \langle cx.CName, cx.City \rangle :$
 $\neg(\exists px(Part) \exists sx(Supply) (cx.C\# = sx.C\# \wedge$
 $sx.P\# = px.P\# \wedge$
 $px.Producer = 'Apple')) \}$
- select CName, City
 from Customer
 where not exists (select *
 from Part, Supply
 where Customer.C# = Supply.C# and
 Supply.P# = Part.P# and
 Producer = 'Apple')

Query 7:

- $\Pi_{CName, City}(\sigma_{Producer = 'Apple'} (Customer \bowtie Supply \bowtie Part)) - \Pi_{CName, City}(\sigma_{Producer \neq 'Apple'} (Customer \bowtie Supply \bowtie Part))$
- $\forall cx(Customer)$
 $\{ \langle cx.CName, cx.City \rangle :$

$$\begin{aligned} & \forall sx(\text{Supply})(sx.C\# = cx.C\# \Rightarrow \\ & \quad \exists px(\text{Part})(sx.P\# = px.P\# \wedge px.Producer = 'Apple')) \wedge \\ & \quad \exists sy(\text{Supply})(sy.C\# = cx.C\#) \} \end{aligned}$$

- select CName, City
from Customer
where 'Apple' = all (select Producer
from Supply, Part
where Customer.C# = Supply.C# and
Supply.P# = Part.P# and
Producer = 'Apple')

Query 8:

- $\Pi_{CName, City, Part\#}(\text{Customer} \bowtie \text{Supply} \bowtie \text{Part}) \div \Pi_{Part\#}(\sigma_{Producer = 'Apple'}(\text{Part}))$
- $\forall cx(\text{Customer})$
 $\{ \langle cx.CName, cx.City \rangle :$
 $\quad \forall px(\text{Part})(px.Producer = 'Apple' \Rightarrow$
 $\quad \quad \exists sx(\text{Supply})(sx.P\# = px.P\# \wedge sx.C\# = cx.C\#)) \}$
- select CName, City
from Customer
where not exists (select *
from Part
where Producer = 'Apple' and
not exists (select *
from Supply
where Supply.P# = Part.P# and
Supply.C# = Customer.C#))

The logic of this SQL expression is: The current customer qualifies if there does not exist an Apple product that is not supplied to the customer.

Query 9:

```
select  AVG(price)
from    Part
where   Producer = 'Apple'
```

Query 10:

```
select  Part.P#, Pname, SUM(Quantity)
from    Part natural join Supply
where   Producer = 'Apple' and
        Date >= '2019-01-01'
group by Part.P#, PName
```

Query 11:

```
select  Part.P#, Pname, sum(Quantity), sum(Amount)
from    Part natural join Supply natural join Customer
where   Producer = 'Apple' and
        City = 'Guelph' and
        Date between '2020-01-01' and '2020-12-31'
group by Part.P#, PName
```

Query 12:

```
select  Part.P#, Pname, sum(Quantity), sum(Amount)
from    Part natural join Supply natural join Customer
where   Producer = 'Apple' and
        City = 'Guelph' and
        Date between '2020-01-01' and '2020-12-31'
group by Part.P#, PName
        having count(distinct Customer.C#) > 10
order by PName
```